

*Моей жене Стелене и трем дочерям, Анне, Люси и Руби,
за терпение и поддержку на протяжении всей работы
над книгой. Спасибо команде разработчиков WinUI
за открытость и готовность к сотрудничеству
с сообществом разработчиков на платформе Windows
по мере развития и отработки WinUI 3.0*

Содержание

От издательства	14
Об авторе	15
О рецензенте	16
Предисловие	17
Часть I. ВВЕДЕНИЕ В ПРИЛОЖЕНИЯ WinUI И WINDOWS	22
Глава 1. Введение в WinUI	23
Технические требования.....	24
До UWP – приложения Windows 8 XAML	24
Дизайн UI-приложения для Windows	25
Windows Runtime (WinRT)	26
Отпор пользователей и путь к Windows 10.....	26
Windows 10 и разработка приложений UWP.....	26
Выбор языка при разработке на платформе UWP	27
Снятие ограничений	28
Обратная совместимость UWP.....	29
Что такое XAML?.....	30
Создание адаптивного UI для любого устройства	32
Привязка к данным	33
Стилизация UI с помощью XAML	36
Отделение представления от бизнес-логики	37
Что такое WinUI?	38
Первый выпуск WinUI.....	39
По дороге к WinUI 3.0.....	41
WinUI 2.1	41
WinUI 2.2	41
WinUI 2.3	42
WinUI 2.4	42
WinUI 2.5	42
Что нового в WinUI 3.0?.....	43
Прощай, UWP?	43
Новые возможности в WinUI 3.0.....	44
Project Reunion и WinUI	45
Сравнение WinUI с другими каркасами разработки для Windows.....	45
WinUI и UWP	45
WinUI и WPF.....	46
Преимущества WinUI	46

Преимущества WPF	46
WinUI и Windows Forms (WinForms).....	47
Преимущества WinUI	47
Преимущества WinForms	48
Резюме	48
Вопросы	48

Глава 2. Конфигурирование среды разработки

и создание проекта	49
Технические требования.....	50
Установка Visual Studio и рабочих нагрузок для разработки в Windows.....	50
Добавление шаблонов WinUI-приложений	51
Первое знакомство с идеей приложения	52
Обзор функций приложения.....	53
Типы проектов WinUI in UWP и WinUI in Desktop	53
Создание первого проекта WinUI.....	54
Анатомия проекта типа WinUI in UWP	56
Обзор файла App.xaml.....	56
Обзор файла App.xaml.cs	57
Обзор файла MainPage.xaml.....	57
Обзор файла MainPage.xaml.cs	58
Обзор ссылок из проекта	58
Обзор свойств проекта.....	59
Основы XAML.....	59
Построение модели	60
Создание демонстрационных данных	62
Построение начальной версии UI	64
Завершение инициализации привязки к данным	65
Создание DataTemplate и привязка к UI.....	66
Еще о WinUI и UWP	68
Модель приложений UWP.....	70
Работа с элементами управления WinUI, свойствами и событиями	71
Добавление заголовка ListView	71
Создание фильтра в виде комбинированного списка	73
Добавление кнопки создания нового предмета	77
Резюме	79
Вопросы	80

Глава 3. MVVM как средство, обеспечивающее удобство сопровождения и тестирования

.....	81
Технические требования.....	82
Что такое MVVM.....	82
MVVM – общая картина	82
Библиотеки MVVM для WinUI	83
Библиотека MVVM из Windows Community Toolkit	84
Библиотека Prism	84

MVVMCross	84
Выбор каркаса для WinUI-приложений.....	85
Устройство привязки к данным в WinUI	85
Что такое расширения разметки?	85
Расширение разметки Binding.....	86
Расширение разметки x:Bind.....	87
Обновление данных представления с помощью INotifyPropertyChanged.....	88
Обновление данных коллекции с помощью INotifyCollectionChanged.....	88
Реализация MVVM в WinUI-приложениях	89
Работа с событиями и командами.....	92
Реализация интерфейса ICommand	93
Использование команд в модели представления.....	93
Обновление представления	95
Выбор каркаса автономного тестирования	97
Резюме	97
Вопросы	98

Глава 4. Дополнительные концепции MVVM

Технические требования.....	99
Основы внедрения зависимостей	100
Использование DI совместно с классами ViewModel.....	101
Использование x:Bind в сочетании с событиями.....	103
Страничная навигация с помощью MVVM и DI	104
Добавление страницы ItemDetailsPage	104
Добавление новых интерфейсов и служб.....	107
Создание службы навигации	107
Создание службы данных	109
Потребление служб упрощает сопровождение	112
Обработка параметров в ItemDetailsPage	114
Создание класса ItemDetailsViewModel.....	115
Резюме	118
Вопросы	118

Глава 5. Элементы управления WinUI

Технические требования.....	119
Что WinUI предлагает разработчикам	120
Анимированный визуальный проигрыватель (Lottie).....	121
Элемент NavigationView.....	121
Элемент ParallaxView	123
Элемент RatingControl.....	124
Элемент TwoPaneView.....	125
Приложение XAML Controls Gallery для Windows	126
Изучаем элемент управления ScrollViewer	127
Что нового в WinUI 3.0.....	129
Обратная совместимость.....	129
Инструментальные средства Visual Studio	129

Контроль входных данных	129
Новый элемент WebView	130
XamlDirect API для разработчиков ПО промежуточного уровня.....	132
Добавление новых элементов управления в проект	134
Использование элемента управления SplitButton.....	134
Добавление элемента TeachingTip к кнопке сохранения	136
Резюме	138
Вопросы	138
Для дальнейшего чтения	139

Глава 6. Использование данных и служб..... 140

Технические требования.....	140
Управление состоянием приложения с помощью событий жизненного цикла	141
События жизненного цикла приложений в Windows	141
События жизненного цикла WinUI-приложений	142
OnLaunched.....	143
OnActivated	143
Выполнение приложений в приоритетном и фоновом режимах	144
Событие Suspending	144
Событие Resuming	145
Создание хранилища данных SQLite	145
Что такое SQLite?	146
Добавление SQLite в DataService	146
Использование Micro ORM для упрощения доступа к данным	149
Добавление Dapper в проект	149
Модификация инициализации службы данных	152
Работа с данными посредством служб.....	154
Контроль правильности данных в MVVM	159
Резюме	161
Вопросы	162

Часть II. РАСШИРЕНИЕ WinUI И МОДЕРНИЗАЦИЯ ПРИЛОЖЕНИЙ..... 163

Глава 7. Система проектирования текучих интерфейсов для приложений Windows..... 164

Технические требования.....	165
Что такое Fluent Design System?.....	165
Изучение текучего дизайна для Windows	166
Элементы управления.....	166
Паттерны.....	166
Макет	168
Ввод.....	169
Стиль.....	169

Включение текучего дизайна в WinUI-приложения.....	171
Изменение полосы заголовка	171
Изменение стиля страницы MainPage.....	172
Изменение стиля страницы ItemDetailsPage.....	176
Использование редактора Fluent XAML Theme Editor.....	178
Цвета.....	180
Формы	181
Использование галереи ресурсов UWP	182
Ресурсы и комплекты инструментов для текучего дизайна	184
Резюме	185
Вопросы.....	185
Глава 8. Построение WinUI-приложений в .NET 5	186
Технические требования.....	187
Создание проекта WinUI в .NET 5	187
Что такое WinUI in Desktop?.....	187
Создание нового проекта типа WinUI in Desktop	189
Структура проекта классического приложения	191
Добавление элемента управления WebView2.....	192
Проект упаковки	194
Визуальные активы в конструкторе манифеста	195
Ссылка на библиотеки .NET 5 из проекта.....	197
Разделение библиотеки .NET 5 с WPF-приложением.....	200
Создание библиотеки элементов управления WinUI	203
Резюме	207
Вопросы.....	208
Глава 9. Улучшение приложений с помощью Windows Community Toolkit	209
Технические требования.....	209
Введение в WCT.....	210
Истоки WCT	210
Обзор последних выпусков комплекта инструментов	211
Изучение демонстрационного приложения Windows Community Toolkit	212
Установка и запуск демонстрационного приложения.....	212
Элементы управления.....	214
Элементы управления WPF и WinForms	217
Использование элементов управления из комплекта инструментов	219
Создание проекта типа WinUI in Desktop	219
Ссылка на пакеты WCT.....	222
Добавление данных в DataGrid.....	223
Добавление элементов управления в MainWindow.....	224
Вспомогательные классы, службы и расширения, входящие в комплект инструментов	226
Вспомогательные классы.....	227
Службы	228

MVVM.....	230
Расширения	230
Резюме	231
Вопросы	232

Глава 10. Модернизация существующих приложений

Win32 с помощью островков XAML	233
Технические требования.....	234
Что такое островки XAML?	234
Модернизация приложения WinForms с помощью островков XAML.....	236
Создание проекта разделяемой библиотеки классов	236
Создание объемлющего проекта WinForms	240
Модернизация WPF-приложения с помощью островков XAML.....	243
Использование элемента управления UWP MapControl в WPF-приложении.....	245
Использование браузерного элемента управления WebViewCompatible в WPF-приложении.....	249
Использование браузерного элемента управления WebView2 в приложении WinForms	251
Резюме	253
Вопросы	253

Часть III. СБОРКА И РАЗВЕРТЫВАНИЕ В WINDOWS, И НЕ ТОЛЬКО

254

Глава 11. Отладка WinUI-приложений в Visual Studio	255
Технические требования.....	256
Отладка в Visual Studio.....	256
Отладка локальных приложений	256
Конструктор XAML	257
Отладка локально установленного приложения	259
Отладка удаленных приложений	261
Типичные ошибки макета XAML.....	264
Ошибки в макете Grid	264
Ошибки применения стиля	264
Улучшение XAML-разметки с помощью статического анализа кода.....	265
Как находить ошибки привязки к данным	267
Типичные ошибки привязки к данным	267
Выбор наилучшего режима привязки.....	268
Генерирование уведомлений PropertyChanged	268
Работа с ObservableCollection<T>	268
Использование окна ошибок привязки XAML	269
ValidateOnBuild проверяет, что для каждого зарегистрированного сервиса зарегистрированы все его зависимости	271

Кодирование с применением горячей перезагрузки XAML.....	272
Отладка с применением динамического дерева визуальных объектов и динамического обозревателя свойств.....	273
Резюме	278
Вопросы	278

Глава 12. Размещение приложения ASP.NET Core Blazor в WinUI

Технические требования.....	279
Приступаем к работе с ASP.NET Core и Blazor.....	280
Немного об истории ASP.NET и ASP.NET Core	280
Что такое Blazor?	281
WebAssembly и клиентская разработка на платформе .NET	282
Создание приложения Blazor Wasm	283
Создание простого приложения для учета задач	286
Варианты развертывания приложений Blazor Wasm	289
Варианты развертывания проектов Blazor Wasm	290
Amazon Web Services (AWS)	290
GitHub Pages	290
Azure App Service	291
Azure Static Web Apps	291
Публикация приложения Blazor на Azure Static Web Apps.....	291
Отправка проекта на GitHub	292
Создание ресурса в службе Azure Static Web Apps	294
Публикация приложения с помощью действий GitHub	296
Размещение приложения Blazor в элементе управления WinUI	
WebView2	299
Резюме	300
Вопросы	300

Глава 13. Сборка, выпуск и мониторинг приложений с помощью Visual Studio App Center

Технические требования.....	302
Приступаем к работе с Visual Studio App Center.....	302
Создание учетной записи в App Center	303
Создание первого приложения в App Center	306
Настройка сборки в App Center	309
Интеграция App Center с репозиторием GitHub	310
Развертывание приложения с помощью App Center	313
Создание ранних релизов приложения для бета-тестеров	314
Мониторинг и сбор аналитических данных о работе приложения	317
Оснащение кода инструментальными средствами.....	317
Сбор и анализ отчетов о сбоях в App Center	320
Резюме	322
Вопросы	322

Глава 14. Упаковка и развертывание WinUI-приложений	323
Технические требования.....	323
Основы упаковки приложений и технологии MSIX.....	324
Что такое MSIX?	324
Обзор инструментов и ресурсов MSIX.....	326
Приступаем к упаковке приложения в Visual Studio	327
Распространение приложений с помощью диспетчера пакетов Windows	331
Добавление пакета в общественный репозиторий	332
Использование WinGet для управления пакетами.....	334
Распространение приложений через Microsoft Store	336
Подготовка бесплатного приложения к отправке в Microsoft Store	336
Загрузка пакета в Store	340
Загрузка WinUI-приложений из сторонних источников с помощью MSIX	343
Создание MSIX-пакета для загрузки из стороннего источника	343
Загрузка MSIX-пакета из стороннего источника	344
Резюме	347
Вопросы	347
Ответы на вопросы	348
Предметный указатель	352

От издательства

Отзывы и пожелания

Мы всегда рады отзывам наших читателей. Расскажите нам, что вы думаете об этой книге – что понравилось или, может быть, не понравилось. Отзывы важны для нас, чтобы выпускать книги, которые будут для вас максимально полезны.

Вы можете написать отзыв на нашем сайте www.dmkpress.com, зайдя на страницу книги и оставив комментарий в разделе «Отзывы и рецензии». Также можно послать письмо главному редактору по адресу dmkpress@gmail.com; при этом укажите название книги в теме письма.

Если вы являетесь экспертом в какой-либо области и заинтересованы в написании новой книги, заполните форму на нашем сайте по адресу http://dmkpress.com/authors/publish_book/ или напишите в издательство по адресу dmkpress@gmail.com.

Скачивание исходного кода примеров

Скачать файлы с дополнительной информацией для книг издательства «ДМК Пресс» можно на сайте www.dmkpress.com на странице с описанием соответствующей книги.

Список опечаток

Хотя мы приняли все возможные меры для того, чтобы обеспечить высокое качество наших текстов, ошибки все равно случаются. Если вы найдете ошибку в одной из наших книг, мы будем очень благодарны, если вы сообщите о ней главному редактору по адресу dmkpress@gmail.com. Сделав это, вы избавите других читателей от недопонимания и поможете нам улучшить последующие издания этой книги.

Нарушение авторских прав

Пиратство в интернете по-прежнему остается насущной проблемой. Издательства «ДМК Пресс» и Packt Publishing очень серьезно относятся к вопросам защиты авторских прав и лицензирования. Если вы столкнетесь в интернете с незаконной публикацией какой-либо из наших книг, пожалуйста, пришлите нам ссылку на интернет-ресурс, чтобы мы могли применить санкции.

Ссылку на подозрительные материалы можно прислать по адресу электронной почты dmkpress@gmail.com.

Мы высоко ценим любую помощь по защите наших авторов, благодаря которой мы можем предоставлять вам качественные материалы.

Об авторе

Элвин Эшкрафт – инженер-программист, лидер технического сообщества с 25-летним стажем разработки ПО. Работает преимущественно с Microsoft Windows, вебом и облачными технологиями, в сферу интересов входит в основном здравоохранение. 11 раз был удостоен звания Microsoft MVP, в последний раз как разработчик Windows.

Элвин работает ведущим инженером-программистом в глобальной компании Allscripts по разработке ПО в области здравоохранения со штаб-квартирой в Филадельфии. Также является членом правления фонда TechBash, где отвечает за организацию ежегодных конференций разработчиков TechBash. Раньше работал в компаниях Oracle, Genzeon, CSC и ITG Pathfinders.

Родился в Аллентайне, штат Пенсильвания, теперь проживает в районе Уэст-Гроув, тоже в Пенсильвании, с женой и тремя детьми.

Хочу поблагодарить всех близких, кто поддерживал меня, особенно свою жену Стелену и трех дочерей.

О рецензенте

Ник Рэндольф в настоящее время возглавляет компанию Built to Roam, специализирующуюся на разработке мобильных приложений с развитым функционалом. Ник удостоен звания Microsoft MVP в знак признания заслуг и за опыт работы на платформах Microsoft.

Ника не раз приглашали на различные мероприятия, в т. ч. Tech Ed и Ignite Australia & NZ, DDD, NDC и в местные группы пользователей. Он автор нескольких книг по разработке для Windows в среде Visual Studio, был членом жюри в финалах международного конкурса Imagine Cup. Ник работал над многочисленными мобильными приложениями и помогал сотням разработчиков создавать свои мобильные приложения. Участвовал в разработке приложений для таких хорошо известных торговых марок, как Domain.com.au, ninemsn, AFL, NRL, Qantas, JB Hi-Fi, NAB, Stan и Boost Juice.

Предисловие

WinUI 3.0 – первый шаг Microsoft в направлении унифицированной платформы разработки Windows. Эта инициатива, получившая название Project Reunion, является попыткой объединить UWP, WPF и другие каркасы для разработки пользовательского интерфейса (UI) на одной платформе. WinUI позволяет быстро создавать приложения Windows, стиль которых адаптируется к платформе. По мере отработки WinUI разработчики получают возможность выбирать в качестве целевой платформы настольные компьютеры под управлением Windows, Xbox, HoloLens, Surface Hub и др.

Разработчики приложений Win32 также могут пользоваться элементами управления WinUI, чтобы модернизировать внешний вид своих приложений с помощью островков XAML из комплекта инструментов Windows Community Toolkit. Этот комплект инструментов с открытым исходным кодом содержит десятки элементов управления и других вспомогательных библиотек для WinUI, UWP и Win32. Вы научитесь выбирать подходящие элементы управления для своих приложений и использовать их в нескольких проектах.

Из этой книги вы узнаете, как разрабатывать, отлаживать, собирать и развертывать приложения с помощью Visual Studio и облачных инструментов из Microsoft Azure и GitHub. Вы откроете для себя, как сделать так, чтобы ваше приложение WinUI могло попасть в руки пользователей как из потребительского, так и из корпоративного сектора. Дочитав книгу до конца, вы будете хорошо понимать, как создавать, модернизировать и распространять приложения Windows на платформе WinUI 3.0.

ПРЕДПОЛАГАЕМАЯ АУДИТОРИЯ

Эта книга адресована всем, кто хочет разрабатывать приложения для Windows 10 с помощью WinUI 3.0. Читатели, знакомые с UWP, WPF и WinForms, тоже смогут почерпнуть из нее много полезного для углубления своих представлений о разработке в Windows и модернизации существующих приложений. Для чтения книги необходимо знакомство с языком C# и платформой .NET, но предварительные знания о разработке интерфейсов с применением WinUI, UWP или XAML не требуются.

СТРУКТУРА КНИГИ

В главе 1 «Введение в WinUI» рассматривается история каркасов UI в Windows и истоки WinUI. Мы создадим свой первый проект WinUI 3.0 в Visual Studio 2019.

В главе 2 «Конфигурирование среды разработки и создание проекта» мы познакомимся с проектом, над которым будем работать на протяжении большей части этой книги. Мы объясним внутреннее устройство проекта WinUI и поговорим об основных особенностях платформы, в т. ч. о жизненном цикле приложения и привязке к данным.

В главе 3 «MVVM как средство, обеспечивающее удобство сопровождения и тестирования» объясняется паттерн проектирования «Модель–представление–представление модели» (MVVM), играющий важнейшую роль в разработке приложений WinUI. Вы также освоите основы автономного тестирования проектов WinUI.

В главе 4 «Дополнительные концепции MVVM» мы продолжим изучение MVVM в WinUI, добавив NuGet-пакет **внедрения зависимостей** (dependency injection – DI) и сервисные классы для страничной навигации и выборки данных.

В главе 5 «Элементы управления WinUI» изучаются элементы управления, имеющиеся в галерее элементов управления XAML для WinUI 3.0. Это приложение с открытым исходным кодом иллюстрирует элементы управления (с примерами), доступные в проектах WinUI. Затем мы добавим два новых элемента из галереи в свой проект.

Глава 6 «Использование данных и служб» продолжает разговор о службах данных, начатый в главе 4, и добавляет хранилище на основе SQLite для сохранения данных нашего приложения. Здесь же мы познакомимся с механизмами контроля, которые препятствуют вводу недопустимых данных пользователями.

В главе 7 «Система проектирования текучих интерфейсов для приложений Windows» рассматривается кросс-платформенная система проектирования текучих интерфейсов от Microsoft. Дается исторический обзор эволюции идей проектирования интерфейсов в Windows. Здесь же мы познакомимся с редактором Fluent XAML Theme Editor, предназначенным для создания собственной темы интерфейса для приложения WinUI.

В главе 8 «Построение приложений WinUI в .NET 5» объясняется, как создать WinUI-интерфейс для проекта классического приложения, ориентированного на платформу .NET 5, а не UWP. Здесь же иллюстрируется создание элемента управления в .NET 5, допускающего использование в нескольких проектах классических WinUI-приложений.

В главе 9 «Улучшение приложений с помощью Windows Community Toolkit» мы познакомимся с комплектом инструментов с открытым исходным кодом Windows Community Toolkit и входящим в него приложением, иллюстрирующим применение Windows.Controls, вспомогательных классов и других библиотек, которые мы добавим в свои проекты.

Глава 10 «Модернизация существующих приложений Win32 с помощью островков XAML» посвящена элементам управления из Windows Community Toolkit на основе технологии островков XAML. Объясняется, как интегрировать их в приложения WPF и WinForms в качестве первого шага на пути к модернизации приложения и полному переходу на WinUI.

В главе 11 «Отладка WinUI-приложений в Visual Studio» мы ближе познакомимся с инструментами отладки XAML, имеющимися в Visual Studio 2019.

Они позволяют разработчикам WinUI оптимизировать и отладить основанный XAML пользовательский интерфейс и код привязки к данным.

В главе 12 «Размещение приложения ASP.NET Core Blazor в WinUI» описывается, как создать и развернуть одностраничное приложение (single-page application – SPA) Blazor в Azure с помощью Visual Studio Code и действий GitHub. Затем развернутое приложение Blazor будет размещено в приложении Windows с помощью нового элемента управления WebView2.

В главе 13 «Сборка, выпуск и мониторинг приложений с помощью Visual Studio App Center» рассматриваются имеющиеся в Visual Studio App Center средства сборки, развертывания и мониторинга приложений Windows. Вы научитесь оснащать свой код инструментальными средствами для получения в реальном времени аналитических данных и данных об аварийных отказах развернутых приложений.

Глава 14 «Упаковка и развертывание WinUI-приложений» посвящена различным методам развертывания WinUI-приложений. Описывается, как использовать для этой цели Visual Studio, Microsoft Partner Center, Microsoft Store и WinGet.

Что необходимо для чтения этой книги

Чтобы получить максимум пользы от чтения книги, необходимо установить Visual Studio 2019 версии 16.9 или более поздней (<https://visualstudio.microsoft.com/downloads/>) со следующими рабочими нагрузками и последней версией NuGet-пакета WinUI 3.0 (<https://marketplace.visualstudio.com/items?itemName=Microsoft-WinUI.WinUIProjectTemplates>):

- Universal Windows Platform Development;
- .NET Desktop Development (включает .NET 5).

Код и инструкции должны работать и для следующих версий рекомендуемого ПО. Самая свежая информация о WinUI 3.0 находится на сайте Microsoft Docs по адресу <https://docs.microsoft.com/en-us/windows/apps/winui/winui3/#installwinui-3-preview-4>.

Программное и аппаратное обеспечение, используемое в книге	Требования к ОС
Visual Studio 2019 версии 16.9	Windows 10 версии 1803 или более поздней
Шаблоны WinUI 3.0	
Visual Studio Code	Windows 10, macOS или Linux

i Поскольку в 2021 году WinUI 3 и Project Reunion все еще пребывают в фазе активной разработки, не исключено, что в будущих версиях имена некоторых упоминаемых в книге проектов, пакетов и библиотек изменятся. Островки XAML поначалу будут недоступны, а UWP-клиенты поддерживаются не полностью. С полным перечнем всего, что планируется включить в первую стабильную версию WinUI 3.0, можно ознакомиться в дорожной карте команды разработчиков на GitHub по адресу <https://github.com/microsoft/microsoft-ui-xaml/blob/master/docs/roadmap.md#winui-30-feature-roadmap>.

Для главы, посвященной веб-разработке с применением Blazor, рекомендуется скачать Visual Studio Code (<https://code.visualstudio.com/>). Если вы еще не установили рабочую нагрузку .NET Desktop Development для Visual Studio, то для разработки с применением Blazor понадобится .NET 5 SDK (<https://dotnet.microsoft.com/download/dotnet/5.0>).

Тем, кто пользуется цифровой версией книги, мы рекомендуем набирать код самостоятельно или скачать его из репозитория на GitHub (ссылка приведена в следующем разделе). Это поможет избежать потенциальных ошибок из-за копирования и вставки кода.

Дополнительные сведения о концепциях разработки для Windows можно почерпнуть из пособия «Develop Windows 10 applications Learning Path» на сайте **Microsoft Learn** (<https://docs.microsoft.com/en-us/learn/paths/develop-windows10-apps/>).

СКАЧИВАНИЕ КОДА ПРИМЕРОВ

Код примеров для этой книги можно скачать, зайдя в свою учетную запись на сайте www.packt.com. Если вы купили книгу где-то еще, то можете зарегистрироваться на странице www.packt.com/support, тогда файлы будут отправлены вам по электронной почте.

Код примеров размещен также на сайте GitHub по адресу <https://github.com/PacktPublishing/Learn-WinUI-3.0>. Все обновления выкладываются в репозиторий на GitHub.

В разделе <https://github.com/PacktPublishing/> есть и другие пакеты кода для нашего обширного каталога книг и видео. Не пропустите!

СКАЧИВАНИЕ ЦВЕТНЫХ ИЗОБРАЖЕНИЙ

Мы также предлагаем PDF-файл, содержащий цветные изображения снимков экрана и рисунков. Его можно скачать по адресу https://static.packt-cdn.com/downloads/9781800208667_ColorImages.pdf.

ОБОЗНАЧЕНИЯ И ГРАФИЧЕСКИЕ ВЫДЕЛЕНИЯ

В этой книге применяется ряд соглашений о наборе текста.

CodeInText: код в тексте, имена таблиц базы данных, папок и файлов, расширения имен файлов, пути к файлам и данные, вводимые пользователем. Например: «Для перехвата события запуска следует переопределить метод `OnLaunched` класса приложения».

Отдельно стоящие фрагменты кода набраны так:

```
while (query.Read())  
{
```



```
var medium = new Medium
{
    Id = query.GetInt32(0),
    Name = query.GetString(1),
    MediaType = (ItemType)query.GetInt32(2)
};
```

Желая привлечь внимание к какой-то части кода, мы выделяем ее полужирным шрифтом, например:

```
private void OnSuspending(object sender, SuspendingEventArgs e)
{
    var deferral = e.SuspendingOperation.GetDeferral();
    deferral.Complete();
}
```

Полужирный: новые термины и важные слова, а также части пользовательского интерфейса. Так выделяются команды меню и текст в диалоговых окнах, например: «После того как приложение начнет выполняться, вы увидите, что новый стиль применен к кнопкам **Submit** и **Cancel**, хотя непосредственно к ним вы никаких стилей не применяли».



Советы выглядят так.



Примечания выглядят так.

ВВЕДЕНИЕ В ПРИЛОЖЕНИЯ WINUI И WINDOWS

WinUI 3.0 – новый каркас UI от Microsoft для разработчиков на платформе Windows. В этой части мы начнем с изучения недавней истории XAML и каркасов UI для Windows, а затем познакомим читателей с WinUI. Для изучения концепций WinUI мы создадим простой проект, а потом добавим в него элементы управления и различные возможности, следуя паттернам проектирования и рекомендованным практикам, в т. ч. MVVM (Model-View-ViewModel). Мы позаботимся о тестопригодности проекта WinUI и воспользуемся механизмом внедрения зависимостей (DI) для включения зависимостей от служб в компоненты приложения.

В эту часть входят следующие главы:

- глава 1 «Введение в WinUI»;
- глава 2 «Конфигурирование среды разработки и создание проекта»;
- глава 3 «MVVM как средство, обеспечивающее удобство сопровождения и тестирования»;
- глава 4 «Дополнительные концепции MVVM»;
- глава 5 «Элементы управления WinUI»;
- глава 6 «Использование данных и служб».

Глава 1

Введение в WinUI

WinUI – это набор элементов управления и библиотек с открытым исходным кодом, предназначенных для использования в приложениях для **универсальной платформы Windows** (Universal Windows Platform – UWP) и Win32. Разработчики для UWP используют **пакет средств разработки (SDK)** и должны указывать версию SDK в свойствах проекта. Выделив элементы управления UWP и компоненты пользовательского интерфейса (**UI**) из Windows SDK и выпустив их в виде библиотек с открытым исходным кодом под названием WinUI, Microsoft смогла сократить цикл разработки по сравнению с самой Windows (поскольку версии Windows SDK привязаны к версиям Windows). Такое разделение позволяет также использовать элементы управления в более старых версиях Windows 10. Хотя в настоящее время рекомендуется строить приложения UWP и Win32 с помощью WinUI, важно понимать место WinUI и UWP в более широкой картине разработки для Windows.

Из этой книги вы узнаете, как создавать приложения для Windows с применением библиотек WinUI 3.0. На протяжении всей книги мы будем разрабатывать реальное приложение, пользуясь рекомендуемыми паттернами и приемами.

Прежде чем приступить к созданию приложения WinUI, важно хорошо ориентироваться в разработке клиентов Windows, в различных типах разметки UI на **расширяемом языке разметки приложений** (Extensible Application Markup Language – XAML) и в отличиях WinUI от других каркасов разработки классических приложений для Windows. Поэтому мы начнем с рассмотрения основ UWP и WinUI.

В этой главе рассматриваются следующие темы:

- что такое UWP и почему Microsoft создала еще один каркас разработки приложений;
- как можно использовать XAML для создания замечательных пользовательских интерфейсов, работающих на устройствах разных размеров и из разных семейств;
- зачем был создан WinUI и как он соотносится с UWP;
- каково место WinUI в общей картине средств разработки для Windows;
- что несет с собой WinUI 3.0.

Не волнуйтесь! Рассмотрение этих основополагающих вещей не займет много времени, зато поместит разработку приложения WinUI в нужный

контекст. Уже в следующей главе мы начнем писать код своего первого проекта WinUI.

ТЕХНИЧЕСКИЕ ТРЕБОВАНИЯ

Для проработки представленных в этой главе примеров потребуется следующее программное обеспечение:

- Windows 10 версии 1803 или более поздней. Узнать версию своей Windows вы можете в разделе **Параметры | Система | О программе**;
- Visual Studio 2019 версии 16.9 или более поздней со следующими рабочими нагрузками: .NET Desktop Development и UWP Development;
- шаблоны проектов WinUI 3.0 – на момент написания книги эти шаблоны можно скачать с сайта Visual Studio Marketplace по адресу <https://marketplace.visualstudio.com/items?itemName=Microsoft-WinUI.WinUIProjectTemplates>. После выпуска WinUI 3.0 шаблоны, вероятно, будут включены в состав Visual Studio.

Исходный код для этой главы доступен на GitHub по адресу <https://github.com/PacktPublishing/-Learn-WinUI-3.0/tree/master/Chapter01>.

i В разделе сайта Microsoft Docs, посвященном WinUI 3.0, имеется обновленное пособие по подготовке рабочей станции к разработке для WinUI: <https://docs.microsoft.com/en-us/uwp/toolkits/winui3/>.

До UWP – приложения Windows 8 XAML

Перед тем как в 2015 году в Windows 10 появились приложения UWP, существовали приложения XAML для Windows 8 и 8.1. Синтаксис XAML и многие **интерфейсы прикладного программирования (API)** не изменились, а сама идея стала очередной попыткой Microsoft добиться универсализации приложений на настольной, мобильной и других платформах (Xbox, дополненная реальность и т. д.). XAML-приложение можно было написать для Windows 8 и Windows Phone. При этом генерировались различные наборы двоичных файлов, которые можно было устанавливать на ПК или Windows Phone.

У этих приложений было много других ограничений, снятых в современных приложениях UWP. Например, они могли работать только в полноэкранном режиме, как показано на рис. 1.1.

Многие другие ограничения приложений для Windows 8 были ослаблены или полностью устранены в приложениях UWP. Эти изменения документированы на рис. 1.2.

Конец ознакомительного фрагмента.

Приобрести книгу можно

в интернет-магазине

«Электронный универс»

e-Univers.ru